Tanta University

Faculty of Engineering

Computer and Automatic Control Department

Numerical Differentiation

Prepared by:

Group 5

2014/2015

Instructor:

Dr. Waheed Kamal Zahra

Physics and Engineering Mathematics Department

Group 5 Coordinator:

Mohamed Magdy Abdel-Gaffar Sallam

Group Members:	Section	Evaluation [10]
1- Ahmed Abdel-Zaher Mohamed Abasery	**1**	
2- Asmaa Mohamed Kamal Nasef	**1**	
3- Hasan Saed Hasan Al-Shennawy	**1**	
4- Sara Hesham Saed Abdel-Wahed	**2**	
5- Abdel-Rahman Asharf Mohamed Shebl	**2**	
6- Amr Ahmed Mohamed Al-Shennawy	**2**	
7- Mohamed Ibrahim Mohamed Seada	**2**	
8- Mohamed Magdy Abdel-Gaffar Sallam	**3**	
9- Mahmoud Khaled Abdel-Aal Al-Shaer	**3**	
10- Hala Nasr Fouad Darraz	**3**	

Contents:

Numerical Differentiation:

In numerical analysis, numerical differentiation describes algorithms for estimating the derivative of a mathematical function or function subroutine using values of the function and perhaps other knowledge about the function.

Suppose an Engineer collects a set of data points relating two properties such as temperature and pressure, or distance and time that and object has fallen or amount of water that it has received.

He might wish to determine on or more derivatives of the function but without an analytic expression for the function, the derivatives can only be approximated numerically.

Approximation for the First Derivative:

Given a function $f\ (x)$, we can obtain the series expansion of $f\ (x)$ about x, assuming that the function has as many continuous derivatives as may be required . Using Taylor's formula

$$f\ (x+h) = f\ (x) + h\, f'(x) + \frac{h^2}{2!}\, f"\ (x)\ + \frac{h^3}{3!}\, f'''\ (x) + \cdots$$

Solving for $f'(x)$, we get

$$f'(x) = \frac{f\ (x+h) - f\ (x)}{h} - \frac{h}{2}\, f''(x) - \frac{h^2}{6}\, f'''(x) + \cdots$$

So that an approximation for $f'(x)$ may be written as

$$\boldsymbol{f'(x) \approx \frac{f\ (x+h) - f\ (x)}{h}}$$

For small value of h .

The above expression is known as the **Forward Difference Approximation.**

Another approximation for $f'(x)$ can be obtained by replacing h with -h above, such that

$$\boldsymbol{f'(x) \approx \frac{f(x) - f\ (x-h)}{h}}$$

This is known as the **Backward Difference Approximation** to $f'(x)$.

Note that the truncation error for both the forward and backward difference approximation to $f'(x)$ is $-\frac{h}{2}\, f''(\rho)$, for some ρ in the interval $(x,\ x+h)$, for the forward difference and for some ρ in $(x-h\ , x)$, for the backward difference .

If we now place $-h$ instead of h in the Taylor series expansion and then subtract the resulting equation from the one above, another form of the approximation to $f'(x)$ is obtained such that

$$f\,(x+h) - f\,(x-h) = f\,(x) - f\,(x) + h\,f'(x) + h\,f'(x) + \frac{h^2}{2!}\,f''(x) - \frac{h^2}{2!}\,f''(x) + \frac{h^3}{3!}\,f'''(x) + \frac{h^3}{3!}\,f'''(x) + \cdots$$

So

$$f'(x) = \frac{f\,(x+h) - f\,(x-h)}{2h} + \frac{h^3}{3}\,f'''(x) - \frac{h^4}{5!}\,f^{(5)} - \cdots$$

Or

$$\boldsymbol{f'(x) \approx \frac{f\,(x+h) - f\,(x-h)}{2h}}$$

for a small value of h .

This is known as the **Central Difference Approximation** of $f'(x)$, which gives us an approximation to $f'(x)$ with an error of the order h^2.

Approximation for the Second Derivative:

Considering now approximations to the second derivative $f''(x)$; consider once more the Taylor series expansion of $f\,(x)$ about x

$$f\,(x+h) = f\,(x) + h\,f'(x) + \frac{h^2}{2!}\,f''(x) + \frac{h^3}{3!}\,f'''(x) + \frac{h^{(4)}}{4!}\,f^{(4)} + \cdots$$

And

$$f\,(x-h\,) = f\,(x) - h\,f'(x) + \frac{h^2}{2!}\,f''(x) - \frac{h^3}{3!}\,f'''(x) + \frac{h^4}{4!}\,f^{(4)} - \cdots$$

Adding the two equation we get

$$f\,(x+h) + f\,(x-h) = 2\,f\,(x) + \frac{2h^2}{2!}\,f''(x) + \frac{2h^4}{4!}\,f^{(4)} + \cdots$$

Now solve for $f''(x)$ to obtain

$$f''(x) = \frac{f\,(x+h) - 2\,f\,(x) + f\,(x-h)}{h^2} - \frac{1}{12}\,h^2\,f^{(4)}\,(x) + \cdots$$

Therefore, an approximation formula for $f''(x)$ is given by

$$\boldsymbol{f''(x) \approx \frac{f\,(x+h) - 2f\,(x) + f\,(x-h)}{h^2}}$$

For a small value of h .

In this case it can see that truncation error is $-\frac{1}{12}\, h^2\, f^{(4)}\,(\rho)$ for some ρ in the interval $(x-h\,, x+h)$.

This formula is known as the **Central Difference Formula** for the second derivative.

Approximation for the Third Derivative:

Considering now approximations to the third derivative $f'''(x)$, Let's start with Taylor series:

$$f(x+\;\; h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)} + \cdots$$

And

$$f(x-\;\; h) = f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) - \frac{h^4}{4!}f^{(4)} + \cdots$$

So the central difference form of first derivative will be

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \cdots$$

In the previous relation, replace every h with $\frac{h}{2}$, so the relation will be

$$f'(x) = \frac{1}{h}\left(f\left(x+\frac{h}{2}\right) - f\left(x+\frac{-h}{2}\right)\right) + \cdots$$

From Taylor series, the second derivative will be

$$f''(x) = \frac{1}{h^2}\left(f\left(x+\frac{2h}{2}\right) - 2f\left(x+\frac{0h}{2}\right) + f\left(x+\frac{-2h}{2}\right)\right) + \cdots$$

The third derivative can be deduced using Taylor expansion:

$$f\left(x+\tfrac{3h}{2}\right) = f(x) + \tfrac{3}{2}hf'(x) + \left(\tfrac{3}{2}\right)^2\tfrac{h^2}{2!}f''(x) + \left(\tfrac{3}{2}\right)^3\tfrac{h^3}{3!}f'''(x) + \cdots \qquad (1)$$

$$f\left(x-\tfrac{3h}{2}\right) = f(x) - \tfrac{3}{2}hf'(x) + \left(\tfrac{3}{2}\right)^2\tfrac{h^2}{2!}f''(x) - \left(\tfrac{3}{2}\right)^3\tfrac{h^3}{3!}f'''(x) + \cdots \qquad (2)$$

$$f\left(x+\tfrac{h}{2}\right) = f(x) + \tfrac{1}{2}hf'(x) + \left(\tfrac{1}{2}\right)^2\tfrac{h^2}{2!}f''(x) + \left(\tfrac{1}{2}\right)^3\tfrac{h^3}{3!}f'''(x) + \cdots \qquad (3)$$

$$f\left(x-\tfrac{h}{2}\right) = f(x) - \tfrac{1}{2}hf'(x) + \left(\tfrac{1}{2}\right)^2\tfrac{h^2}{2!}f''(x) - \left(\tfrac{1}{2}\right)^3\tfrac{h^3}{3!}f'''(x) + \cdots \qquad (4)$$

Subtracting (2) from (1) we get

$$f\left(x+\frac{3h}{2}\right)-f\left(x-\frac{3h}{2}\right)=2\left(\frac{3}{2}hf'(x)+\left(\frac{3}{2}\right)^3\frac{h^3}{3!}f'''(x)\right)+\cdots \qquad \textit{(a)}$$

Subtracting (4) from (3) we get

$$f\left(x+\frac{h}{2}\right)-f\left(x-\frac{h}{2}\right)=2\left(\frac{1}{2}hf'(x)+\left(\frac{1}{2}\right)^3\frac{h^3}{3!}f'''(x)\right)+\cdots \qquad \textit{(b)}$$

Multiply (a) by 3 and add it to (b), we get

$$f\left(x+\frac{3h}{2}\right)-f\left(x-\frac{3h}{2}\right)-3\left(f\left(x+\frac{h}{2}\right)-f\left(x-\frac{h}{2}\right)\right)=2\left(\left(\frac{3}{2}\right)^3\frac{h^3}{3!}f'''(x)-3\left(\left(\frac{1}{2}\right)^3\frac{h^3}{3!}f'''(x)\right)\right)+\cdots$$

It can be written as

$$f\left(x+\frac{3h}{2}\right)-f\left(x-\frac{3h}{2}\right)-3\left(f\left(x+\frac{h}{2}\right)-f\left(x-\frac{h}{2}\right)\right)=h^3f'''(x)+\cdots$$

So the third derivative will be:

$$\boldsymbol{f'''(x)\ \approx\frac{1}{h^3}\left(f\left(x+\frac{3h}{2}\right)-3f\left(x+\frac{h}{2}\right)+3f\left(x-\frac{h}{2}\right)-f\left(x+\frac{3h}{2}\right)\right)}$$

With a truncation error $\mathrm{O}(\mathrm{h}^2)$

Approximation for Higher Order Derivatives:

It's obvious that from the formula of f', $f''$and f''' that

$$\boldsymbol{f^{(n)}(x)=\frac{1}{h^n}*\sum_{i=0}^{n}\left[(-1)^i*\binom{n}{i}*f\left(x+h\left(\frac{n}{2}-i\right)\right)\right]+T.E.} \qquad \textbf{\textit{(N)}}$$

Where T.E. is the truncation error always being $\mathrm{O}(\mathrm{h}^2)$

Using this relation we can find that:

$$f^{(4)}(x)=\frac{1}{h^4}\big(f(x+2h)-4f(x+h)+6f(x)-4f(x-h)+f(x-2h)\big)+T.E.$$

Using Taylor Series get the fourth derivative $f^{(4)}$ formula and thus proving the relation (N)

$$f(x+2h) = f(x) + 2hf'(x) + 2^2\frac{h^2}{2!}f''(x) + 2^3\frac{h^3}{3!}f'''(x) + 2^4\frac{h^4}{4!}f^{(4)} + \cdots \tag{5}$$

$$-4f(x+h) = -4\left(f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)} + \cdots\right) \tag{6}$$

$$-4f(x-\ h) = -4\left(f(x) - hf'(x) + \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) - \frac{h^4}{4!}f^{(4)} + \cdots\right) \tag{7}$$

$$f(x-2h) = f(x) - 2hf'(x) + 2^2\frac{h^2}{2!}f''(x) - 2^3\frac{h^3}{3!}f'''(x) + 2^4\frac{h^4}{4!}f^{(4)} + \cdots \tag{8}$$

By the addition of (5), (6), (7), (8) and $\boldsymbol{6f(x)}$ we get :

$$C.O.f(x) = [1 - 4 + \mathbf{6} - 4 + 1] = 0$$

$$C.O.hf'(x) = \ [2 - 4 + 4 - 2] = 0$$

$$C.O.\frac{h^2}{2!}f''(x) = [2^2 - 4 - 4 + 2^2] = 0$$

$$C.O.\frac{h^3}{3!}f'''(x) = [2^3 - 4 + 4 - 2^3] = 0$$

$$C.O.\frac{h^4}{4!}f^{(4)}(x) = [2^4 - 4 - 4 + 2^4] = 24$$

$$\boldsymbol{f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h) \simeq 24 * \frac{h^4}{4!}f^{(4)}(x)}$$

Hence, the fourth derivative $f^{(4)}$ is

$$\boldsymbol{f^{(4)}(x) = \frac{1}{h^4}\big(f(x+2h) - 4f(x+h) + 6f(x) - 4f(x-h) + f(x-2h)\big) + T.E.}$$

The same idea is applicable to prove (N) for $\mathrm{f}^{(5)}$ and $\mathrm{f}^{(6)}$ and so on.

$$\boldsymbol{f^{(5)}(x) = \frac{1}{h^5}\left(f\left(x+\frac{5}{2}h\right) - 5f\left(x+\frac{3}{2}h\right) + 10f\left(x+\frac{1}{2}h\right) - 10f\left(x-\frac{1}{2}h\right) + \ 5f\left(x-\frac{3}{2}h\right) - f\left(x-\frac{5}{2}h\right)\right)}$$

$$\boldsymbol{f^{(6)}(x) = \frac{1}{h^6}\big(f(x+3h) - 6f(x+2h) + 15f(x+h) - 20f(x) + 15f(x-h) - 6f(x-2h) + f(x-3h)\big)}$$

Richardson's Extrapolation:

The central difference approximation gives a truncation error $\mathrm{TE} = O(h^2)$, while the forward difference approximation gives a truncation error $\mathrm{T.E.} = O(h\ )$.

We can reduce the $\mathrm{T.E.}$ more than $O(h^2)$ by technique known as "Richardson extrapolation".

Richardson's extrapolation is an additional technique for approximating derivatives of a function f that will enable us to reduce the truncation error.

We therefore start with the Taylor expansions of $f(x \pm h)$:

$$f(x+h) = f\ (x) + hf'(x) + \frac{h^2}{2!}\ f''(x) + \frac{h^3}{3!}\ f'''(x) + \cdots$$

$$f(x-h) = f\ (x) - hf'(x) + \frac{h^2}{2!}\ f''(x) - \frac{h^3}{3!}\ f'''(x) + \cdots$$

Hence the central difference approximation:

$$\boldsymbol{f'(x) = \frac{f(x+h)-f(x-h)}{2h} + \frac{h^2}{3!} f^{(3)}(x) + \frac{h^3}{4!}\ f^{(4)}(x) + \cdots} \qquad \textit{(9)}$$

Rewrite (9) as:

$$\boldsymbol{L = D(h) + c_2\ h^2 + c_4\ h^4 + \cdots} \qquad \textit{(10)}$$

Where L denotes the quantity that we are interested in approximating

$$\boldsymbol{L = f'(x)}$$

And $D(h)$is the approximation, which in this case is

$$\boldsymbol{D(h) = \frac{f(x+h) - f(x-h)}{2h}}$$

The error is

$$\boldsymbol{E = \ c_2\ h^2 + c_4\ h^4 + \cdots,}$$

Where c_i denotes the coefficient of h_i in (9).

By trying to eliminate the term $c_2 h^2$ the truncation error will be improved.

If we replace h in equation (10) with $\frac{h}{2}$

$$\boldsymbol{f'(x) = D\left(\frac{h}{2}\right) + \ c_2\ \left(\frac{h}{2}\right)^2 + c_4\ \left(\frac{h}{2}\right)^4 + c_6\ \left(\frac{h}{2}\right)^6 + \cdots} \qquad \textit{(11)}$$

The terms in h^2 can be eliminated by (4 * (11) – (10)). We get

$$3f'(x) = 4D\left(\frac{h}{2}\right) - D(h) + \ 4c_2\left(\frac{h}{2}\right)^2 - c_2\,h^2 + 4c_4\left(\frac{h}{2}\right)^4 - c_4\,h^4 + \cdots$$

And it reduces to

$$\mathbf{3f'(x) = \frac{4}{3}D\left(\frac{h}{2}\right) - \ \frac{1}{3}D(h) - \frac{1}{4}c_4\,h^4 + \cdots}$$

The accuracy has been improved by having T.E. of $O(h^4)$.

Thus, we get

$$\boldsymbol{f'(x) \approx \frac{4}{3}\,D\left(\frac{h}{2}\right) - \frac{1}{3}\,D(h)}$$

$$\boldsymbol{TE \ \leq \frac{h^4}{5! * 4}\,f^{(5)}(\rho)}$$

Where $f^{(5)}(\rho)$ the greatest value of the fifth derivative on interval $[x-h\ ,\ x+h\,]$ and

$$\boldsymbol{D\left(\frac{h}{2}\right) = \frac{f\left(x+\frac{h}{2}\right) - f\left(x-\frac{h}{2}\right)}{h}\ ,\ D(h) = \frac{f\,(x+h\,) - f\,(\,x-h\,)}{2h}}$$

This extrapolation method can be repeated to get higher accuracy by having a $T.E. = O(h^4)$, $T.E. = O(h^6)$ and so on.

In general, given and approximation $D(h)$ and having computed the values

$$d_{i,1} = D\left(\frac{h}{2^{i-1}}\right), \qquad i = 1, 2, \ldots$$

For a given $h > 0$, the process can be extended to recursively generate j columns by the formula

$$d_{i,j+1} = \frac{4^j}{4^j - 1}\,d_{i,j} - \frac{1}{4^j - 1}\,d_{i-1,j}$$

The Truncation Error associated with the entry $d_{i,j+1}$ is of order $O(h^{2j+2})$. By arranging the quantities in a tabular form, the procedure can be better illustrated

i,j	h	$d_{i,1}, O(h^2)$	$d_{i,2}, O(h^4)$	$d_{i,3}, O(h^6)$	$\cdots$	$d_{i,j+1}, O(h^{2j+2})$
1	h	$d_{1,1}$	---	---	---	---
2	$\frac{h}{2}$	$d_{2,1}$	$d_{2,2}$	---	---	---
3	$\frac{h}{4}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$	---	---
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	---
i	$\frac{h}{(2^{i-1})}$	$d_{i,1}$	$d_{i,2}$	$d_{i,3}$	$\cdots$	$d_{i,i}$

Examples:

1- Consider the function $(x) = e^x \sin(x)$. Find the fourth derivative $f^{(4)}(2.2)$.

<u>**Solution**</u>

Using $h = 0.1$ and MATLAB program (first_sixth_derivative.m)

```
+---------------+------------+-------------------+
|       x       |     h      |   4 derivative    |
+---------------+------------+-------------------+
|       2.2000  |    0.1000  |      -29.1158     |
+---------------+------------+-------------------+
```

$\therefore f^{(4)}(2.2) = -29.1158$

2- Given the data in the following table:

t	0	0.5	1.0	1.5	2.0
y	0	0.19	0.26	0.29	0.31

Give the central difference approximations for $f''(1), f'''(1), f^{(4)}(1)$.

<u>**Solution**</u>

Using $h = 0.5$ and MATLAB program (first_sixth_derivative.m)

- For $f''(x)$

```
+---------------+---------------+--------------------+
|       x       |       y       |    2 derivative    |
+---------------+---------------+--------------------+
|       0.0000  |       0.0000  | Can`t be calculated |
|       0.5000  |       0.1900  |         -0.4800    |
|       1.0000  |       0.2600  |         -0.1600    |
|       1.5000  |       0.2900  |         -0.0400    |
|       2.0000  |       0.3100  | Can`t be calculated |
+---------------+---------------+--------------------+
```

$\therefore f''(1) = -0.1600$

- For $f'''(x)$

```
+---------------+---------------+---------------------+
|       x       |       y       |     3 derivative    |
+---------------+---------------+---------------------+
|       0.0000  |       0.0000  | Can`t be calculated |
|       0.5000  |       0.1900  | Can`t be calculated |
|       1.0000  |       0.2600  |          0.4400     |
|       1.5000  |       0.2900  | Can`t be calculated |
|       2.0000  |       0.3100  | Can`t be calculated |
+---------------+---------------+---------------------+
```

$\therefore \boldsymbol{f'''(1) = \ 0.4400}$

- For $f^{(4)}(x)$

```
+---------------+---------------+---------------------+
|       x       |       y       |     4 derivative    |
+---------------+---------------+---------------------+
|       0.0000  |       0.0000  | Can`t be calculated |
|       0.5000  |       0.1900  | Can`t be calculated |
|       1.0000  |       0.2300  |         -1.7600     |
|       1.5000  |       0.2600  | Can`t be calculated |
|       2.0000  |       0.3100  | Can`t be calculated |
+---------------+---------------+---------------------+
```

$\therefore \boldsymbol{f^{(4)}(1) = \ -1.7600}$

3- Given the evenly spaced data points, Compute $\boldsymbol{f'(0)}$

$\boldsymbol{x}$	**0**	**0.1**	**0.2**	**0.3**	**0.4**
$\boldsymbol{f(x)}$	**0.0000**	**0.0819**	**0.1341**	**0.1646**	**0.1797**

Solution

Using MATLAB program (first_deriv_forward.m)

```
+---------------+---------------+---------------------+
|       x       |       y       |   first derivative  |
+---------------+---------------+---------------------+
|       0.0000  |       0.0000  |          0.8190     |
|       0.1000  |       0.0819  |          0.5220     |
|       0.2000  |       0.1341  |          0.3050     |
|       0.3000  |       0.1646  |          0.1510     |
|       0.4000  |       0.1797  | Can`t be calculated |
+---------------+---------------+---------------------+
```

$\therefore \boldsymbol{f'(0) = 0.8190}$

4- **In a circuit with impressed voltage $\varepsilon(t) = L\frac{di}{dt} + Ri$ where:**
R is the resistance in the circuit and i is the current. Suppose we measure the current for several values of t and obtain:

t	1.00	1.01	1.02	1.03	1.04
i	3.10	3.12	3.14	3.18	3.24

Where t is measured in seconds, i is in amperes, the inductance L is a constant 0.98 henries, and the resistance in 0.142 ohms. Approximate the voltage $\varepsilon(t)$ when $t = 1.00, 1.01, 1.02, 1.03$ and $.04$.

Solution

Using MATLAB program (first_deriv_forward.m) to get $\frac{di}{dt}$ for the first value of t, MATLAB program (first_deriv_backward.m) to get $\frac{di}{dt}$ for the last value of t, and MATLAB program (first_deriv_central.m) to get $\frac{di}{dt}$ for the remaining value of t

```
+---------------+---------------+---------------------+
|       x       |       y       |   first derivative  |
+---------------+---------------+---------------------+
|       1.0000  |       3.1000  |          2.0000     |
|       1.0100  |       3.1200  |          2.0000     |
|       1.0200  |       3.1400  |          3.0000     |
|       1.0300  |       3.1800  |          5.0000     |
|       1.0400  |       3.2400  |          6.0000     |
+---------------+---------------+---------------------+
```

Since $= 0.98\ H$, $R = 0.142\ ohm$

$$\boldsymbol{\varepsilon(1.00) = (0.98)(2.00) + (0.142\,)(3.10) = 2.40702\,V} \qquad \textbf{(1)}$$
$$\boldsymbol{\varepsilon(1.01) = (0.98)(2.00) + (0.142\,)(3.12) = 2.40304\,V} \qquad \textbf{(2)}$$
$$\boldsymbol{\varepsilon(1.02) = (0.98)(3.00) + (0.142\,)(3.14) = 3.38588\,V} \qquad \textbf{(3)}$$
$$\boldsymbol{\varepsilon(1.03) = (0.98)(5.00) + (0.142\,)(3.18) = 5.33156\,V} \qquad \textbf{(4)}$$
$$\boldsymbol{\varepsilon(1.04) = (0.98)(6.00) + (0.142\,)(3.24) = 6.34008\,V} \qquad \textbf{(5)}$$

5- **Find approximation to $f'(1.8)$ for $f(x) = \ln(x)$ with $h = 0.05$. Then use Richardson's Extrapolation on these values to see if this results in a better approximation.**

Solution

Using MATLAB program (first_deriv_forward.m) to find $f'(1.8)$ with T.E. = $O(h)$

x	h	first derivative
1.8000	0.0500	0.5480

Using MATLAB program (richardson.m) to use Richardson's Extrapolation with accuracy $O(h^2)$, $O(h^4)$, $O(h^6)$ and $O(h^8)$

i	h	Di,1	Di,2	Di,3	Di,4
1	0.0500	0.5557			
2	0.0250	0.5556	0.5556		
3	0.0125	0.5556	0.5556	0.5556	
4	0.0063	0.5556	0.5556	0.5556	0.5556

Where $f'(1.8) = 5555555556$

We see that Richardson's Extrapolation results in a better accuracy lowering the Truncation Error. But, it is not very obvious for higher $O(h)$ because of the Round-Off error.

6- Taylor Theorem can be used to show that centered-difference formula to approximate

$$\boldsymbol{f'(x_0) = \frac{1}{h}[\,f(x_0+h) - \ f(x_0-h)] - \frac{h^2}{6}f'''(x_0) - \ \frac{h^5}{120}f^{(5)}(x_0) - \cdots}$$

Find approximation with a T.E. of $\boldsymbol{O(h^2), O(h^4)}$ and $\boldsymbol{O(h^6)}$ for $\boldsymbol{f'(2.0)}$ when $\boldsymbol{f(x) = xe^x}$ and $\boldsymbol{= 0.2}$.

Solution

Using MATLAB program (richardson.m) to use Richardson's Extrapolation with accuracy $O(h^2)$, $O(h^4)$ and $O(h^6)$

i	h	Di,1	Di,2	Di,3
1	0.2000	22.4142		
2	0.1000	22.2288	22.1670	
3	0.0500	22.1826	22.1672	22.1672

- Having a T.E. of $O(h^2)$ results in $f'(2.0)$ = 22.4142
- Having a T.E. of $O(h^4)$ results in $f'(2.0)$ = 22.1670
- Having a T.E. of $O(h^6)$ results in $f'(2.0)$ = 22.1672

Note that having higher $O(h)$ results in a better accuracy.

7- The time t [s] for a body to move from distance 40 meters to a distance d [m] is given below for five values of d :

t [s]	0.00	0.50	1.00	1.50	2.00
d [m]	40	43	60	77.5	89

Find the speed v and acceleration a at $t = 1\ s$

Solution

Since $v = \frac{dd}{dt}$, Using MATLAB program (first_deriv_central.m) to find v (1.00) gives:

```
+---------------+---------------+---------------------+
|       x       |       y       |   first derivative  |
+---------------+---------------+---------------------+
|        0.0000 |       40.0000 | Can`t be calculated |
|        0.5000 |       43.0000 |          20.0000    |
|        1.0000 |       60.0000 |          34.5000    |
|        1.5000 |       77.5000 |          29.0000    |
|        2.0000 |       89.0000 | Can`t be calculated |
+---------------+---------------+---------------------+
```

$\therefore \boldsymbol{v\ (1.00) = 34.5\ m/s}$

Since $a = \frac{d^2d}{dt^2}$, Using MATLAB program (first_sixth_derivative.m) to find a (1.00) gives:

```
+---------------+---------------+---------------------+
|       x       |       y       |     2 derivative    |
+---------------+---------------+---------------------+
|        0.0000 |       40.0000 | Can`t be calculated |
|        0.5000 |       43.0000 |          56.0000    |
|        1.0000 |       60.0000 |           2.0000    |
|        1.5000 |       77.5000 |         -24.0000    |
|        2.0000 |       89.0000 | Can`t be calculated |
+---------------+---------------+---------------------+
```

$\therefore \boldsymbol{a\ (1.00) = 2.0\ m/s^2}$

8- The altitude h [m] attained by an aircraft climbing from sea level at its maximum rate of climb [m/s] is shown at four times t [min]:

t [min]	10	20	30	40
h [m]	5400	9000	11600	12800

Estimate the aircraft's rate of climb ($\frac{dh}{dt}$) at altitude $h = 11600$ m .

Solution

Using MATLAB program (first_deriv_central.m) to calculate $h'(11600)$gives

```
+---------------+---------------+---------------------+
|       x       |       y       |   first derivative  |
+---------------+---------------+---------------------+
|      10.0000  |     5400.0000 | Can`t be calculated |
|      20.0000  |     9000.0000 |        310.0000     |
|      30.0000  |    11600.0000 |        190.0000     |
|      40.0000  |    12800.0000 | Can`t be calculated |
+---------------+---------------+---------------------+
```

Hence, aircraft's rate of climb h' at $\boldsymbol{h = 11600\ m}$ is $\boldsymbol{190\ m/s}$.

9- A rod is rotating in a plane. The following table gives the angle θ [rad] through which the rod has turned for various values of time t [s]. Calculate the angular velocity ω and angular acceleration α of the rod at $t = 0.6$ s.

t	0	0.2	0.4	0.6	0.8	1.0	1.2
θ	0	0.12	0.49	1.12	2.02	3.20	4.67

Solution

Using MATLAB program (first_sixth_derivative.m) to calculate $\omega\ (0.6) = \frac{d\theta}{dt}(0.6)$

```
+---------------+---------------+---------------------+
|       x       |       y       |     1 derivative    |
+---------------+---------------+---------------------+
|       0.0000  |        0.0000 | Can`t be calculated |
|       0.2000  |        0.1200 |          1.2250     |
|       0.4000  |        0.4900 |          2.5000     |
|       0.6000  |        1.1200 |          3.8250     |
|       0.8000  |        2.0200 |          5.2000     |
|       1.0000  |        3.2000 |          6.6250     |
|       1.2000  |        4.6700 | Can`t be calculated |
+---------------+---------------+---------------------+
```

$\therefore\ \boldsymbol{\omega\ (0.6) = 3.8250\ rad/s}$

Using MATLAB program (first_sixth_derivative.m) to calculate $\alpha\,(0.6) = \frac{d^2\theta}{dt^2}(0.6)$

x	y	2 derivative
0.0000	0.0000	Can`t be calculated
0.2000	0.1200	6.2500
0.4000	0.4900	6.5000
0.6000	1.1200	6.7500
0.8000	2.0200	7.0000
1.0000	3.2000	7.2500
1.2000	4.6700	Can`t be calculated

$\therefore\ \boldsymbol{\alpha\,(0.6) = 3.8250\, rad/s^2}$

10- Given $(x) = \frac{\sqrt{x^3}}{ln(x^4)} + tan(e^{3x})$. Approximate $f^{(5)}(x)$ and $f^{(6)}(x)$ from $x = 0.5$ to $x = 0.8$ with $h = 0.02$.

Solution

Using MATLAB program (n_derivative.m) to for $f^{(5)}(x)$ gives

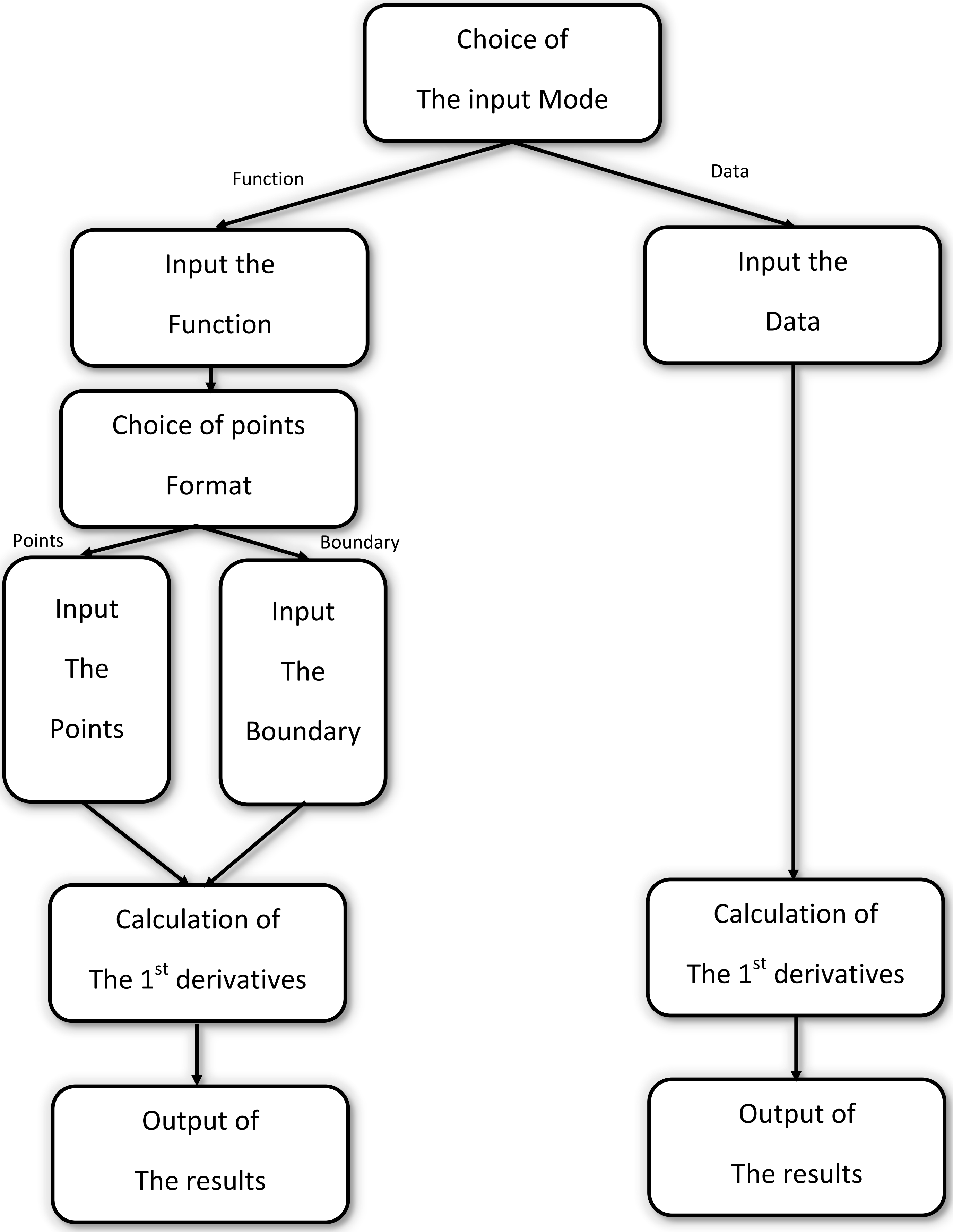
x	h	5 derivative
0.5000	0.0200	34267433518.8024
0.5200	0.0200	-42927896685.2039
0.5400	0.0200	27380868679.6787
0.5600	0.0200	-7845905514.1852
0.5800	0.0200	471174460.9699
0.6000	0.0200	66660390.1552
0.6200	0.0200	172608783.1919
0.6400	0.0200	-6345249309.2422
0.6600	0.0200	26786279211.6589
0.6800	0.0200	-47972667422.2166
0.7000	0.0200	43421810751.8765
0.7200	0.0200	-19329788330.8946
0.7400	0.0200	3608359902.9256
0.7600	0.0200	-4064180262.5284
0.7800	0.0200	11995601850.9831
0.8000	0.0200	-16280002599.5907

Using MATLAB program (n_derivative.m) to for $f^{(6)}(x)$ gives

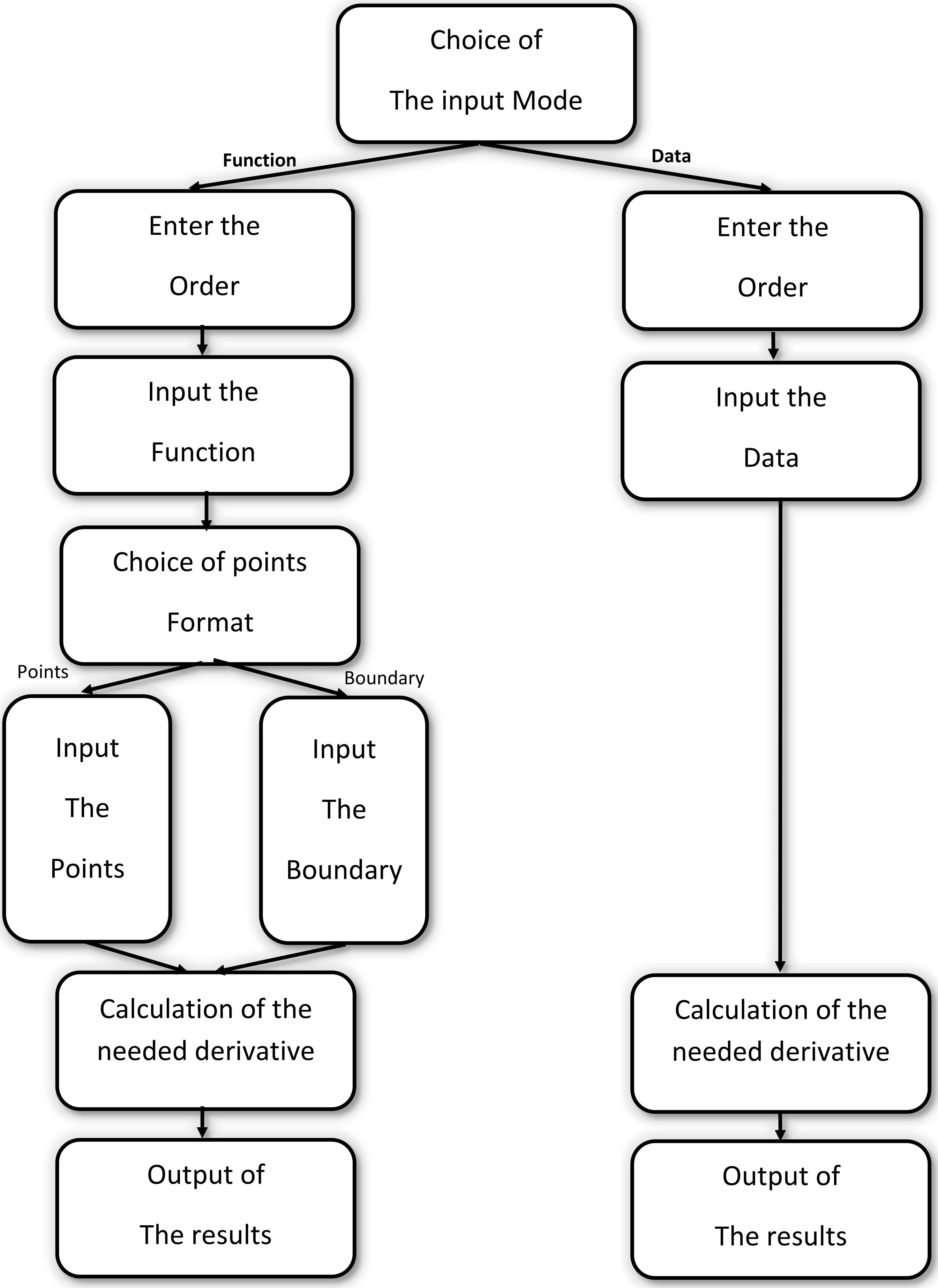
x	h	6 derivative
0.5000	0.0200	-5785324988021.5986
0.5200	0.0200	7016442104430.0547
0.5400	0.0200	-4780347384695.6123
0.5600	0.0200	1700422929929.8767
0.5800	0.0200	-230458042287.5795
0.6000	0.0200	-3196470565.1838
0.6200	0.0200	42610190635.8265
0.6400	0.0200	-423968503056.7980
0.6600	0.0200	1374495883216.5879
0.6800	0.0200	-2256680341100.7251
0.7000	0.0200	2072128797821.9556
0.7200	0.0200	-1043431143895.0142
0.7400	0.0200	-378414453515.0197
0.7600	0.0200	3582703543645.4082
0.7800	0.0200	-8732886598788.3535
0.8000	0.0200	11364408063333.9940

How MATLAB Programs Work:

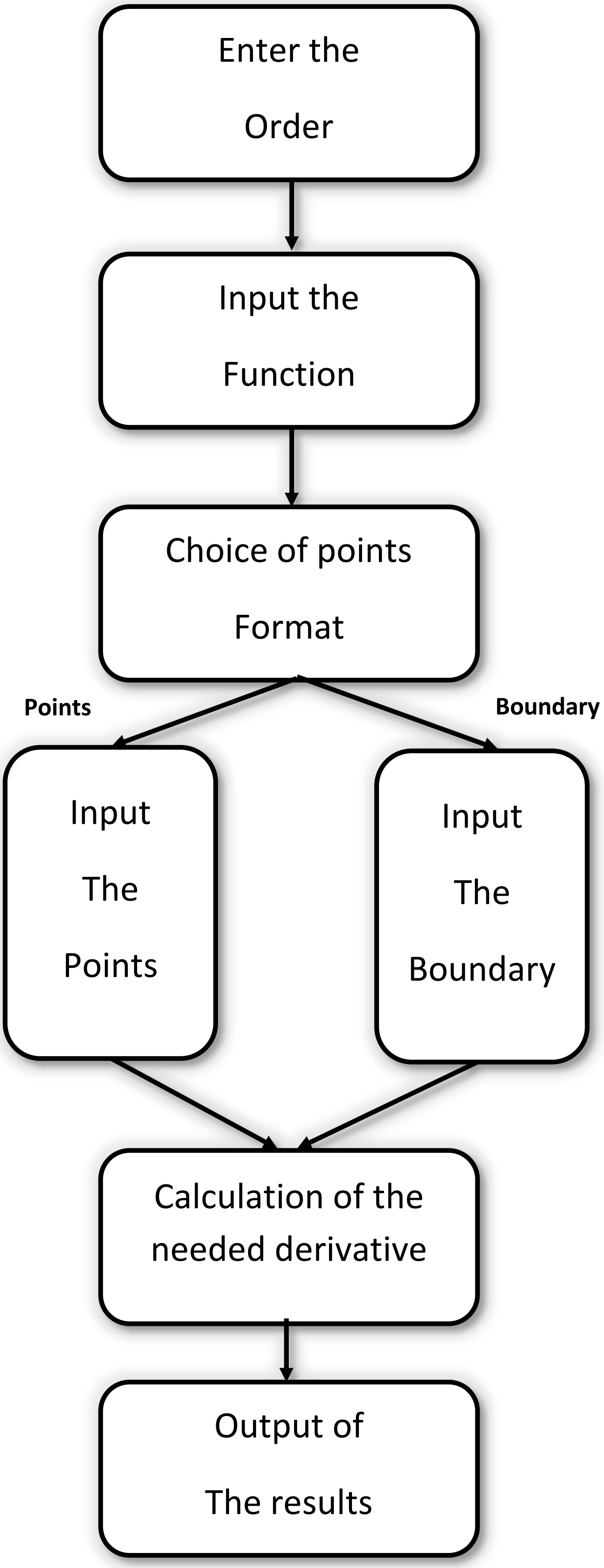
<u>Backward (Forward, Central point) Method:</u>



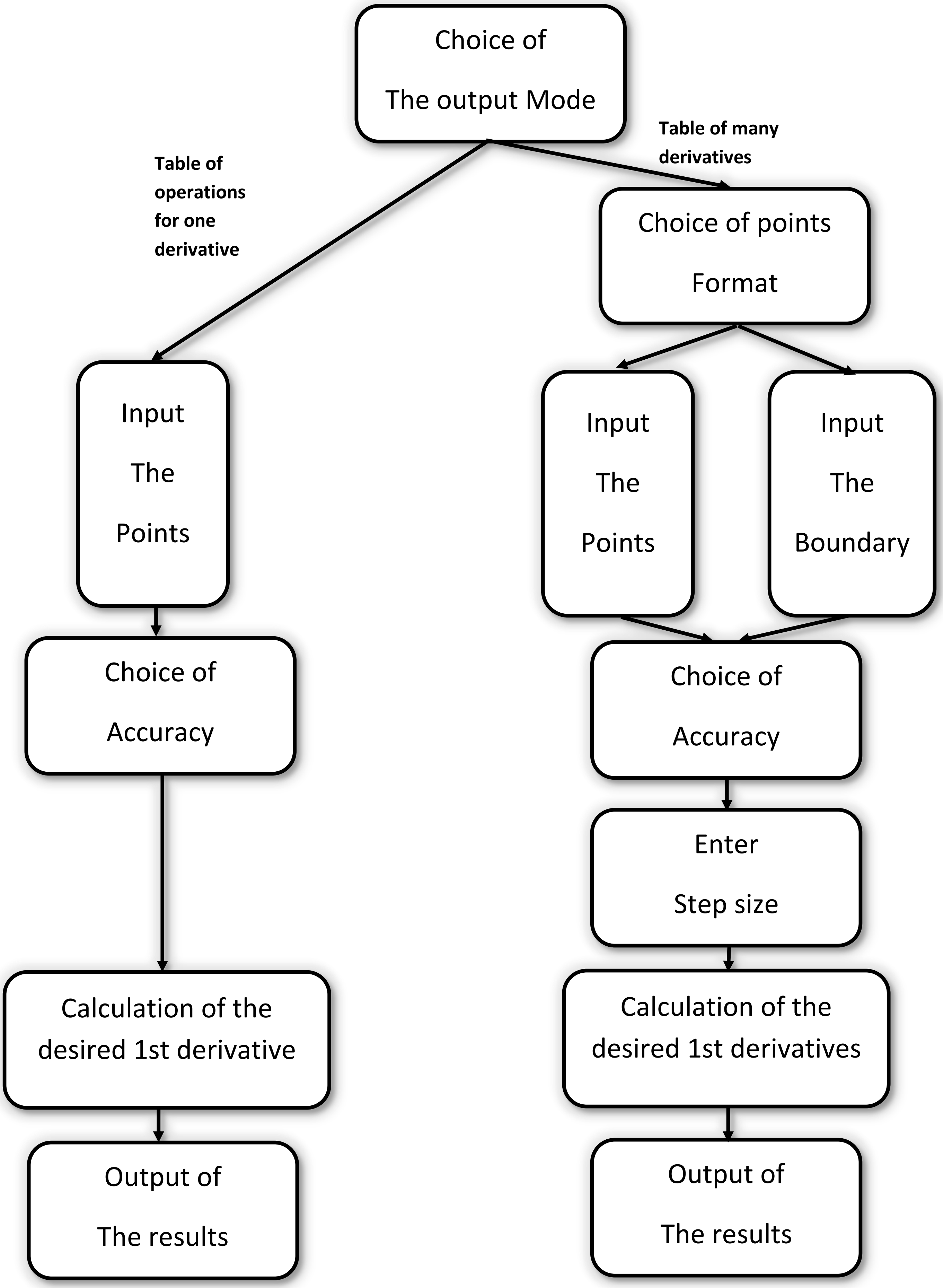
Higher order up to the 6th order:



Higher Order Derivatives:



Richardson's Extrapolation:



Used MATLAB Functions:

1- **input (**parameter1, parameter2**)**: Takes the input from the user
parameter1: a **string** parameter. It outputs a massage to tell the user what to input.
Escape character like "\n", "\\" can be used.
parameter2: a **character** parameter. It is used to when having a **string** input rather than **expression**.

2- **inline (**parameter**)**: Converts string to a function
parameter: a **string** parameter that stores the function to a string in order to **inline** it by the **inline()** function.

3- **disp (**parameter**):** Prints the output on the screen.
parameter: this can be **any type** parameter like **variable** or **string.**
The function **disp()** ends the line after execution.
You cannot use escape characters like "\n".

4- **strcmp (**string1, string2**)**: Compares two strings and returns logical '1' if they are identical, and logical '0' if they are not.
string1 & string2: a **string** parameters.
The function **strcmp()** is case insensitive.

5- **abs (**parameter1**)**: Returns the absolute value of "paramater1".

6- **int32 (**x**)**: Approximates the value of 'x' to the nearest **integer** number.

7- **numel (**x**)**: Returns the number of elements in 'x'.

8- **fprintf (**format, x, y, …**)**: Format and print the output just like in C++

9- **double (**x**)**: Convert the type of variable 'x' to **double.**

10- **strcat (**string1, string2**)**: Returns the concatenation of **string1** and **string2**.
Example:
string1 = 'TEST1'
string2 = 'TEST2'
TEST3 = **strcat (**string1, string2**)** → TEST3 = 'TEST1TEST2'

11- **feval (**parameter, arg1, arg2, arg3, …**)**: Evaluates the value of “parameter” that has **inlined** content using “parameter” arguments arg1, arg2, arg3 … .

Example: f = **inline (**'b/a'**)** → f(a,b) = b/a

feval (f, 5, 2**)** → f = 2/5 = 0.4

12- **floor (**parameter**):** Approximates the value of “parameter” to the first lower integer number.

Note: Since **MATLAB R2013a** warns about removing **inline()** function in its future release, here is **another method** for inserting a function in **MATLAB without using inline() and feval() functions.**

```
x = sym('x');                  % Define a symbol x
f = input('Enter f(x)= ');     % Enter an "expression" for f(x)
                               % not "string" as in inline
f = symfun(f,x);               % Convert sym to sym function
result = f(5);                 % Evaluate the function for x = 5
result = double(result)        % Convert result to double
```

First Derivative-Central Difference:

```
1.  %% Central Difference
2.  % The first derivative " *Central Difference* " metrhod is implemented as
3.  % shown here using MATLAB(R).
4.
5.  %% Clear Memory and Screen
6.  clear
7.  clc
8.
9.  %% Firstly, the user has to choose between two input modes
10. % 1- Function: Formula of a function.
11. % 2- Data    : some (x,y) points.
12. % the user has to write one of two words (Function or Data) to indicate the mode.
13. while 1
14.     disp('Please Choose your Input mode (Function or Data)');
15.     Mode = input('','s');
16.     disp('-------------------------------------');
17.     if ~(strcmpi( Mode , 'Function' )||strcmpi( Mode , 'Data' ))
18.         disp('Error!')
19.     else
20.         break;
21.     end;
22. end;
23.
24. %% Function Mode
25. % The user has chosen the Function mode (Go to line 10 ).
26. if strcmpi( Mode , 'Function' )
27.     fs = input('Please Enter The Function \n f(x)= ','s');
28.     f  = inline(fs);
29.     disp('-------------------------------------');
30.
31.     % Here, the user has to define where he wants to get the values of the 1st derivative.
32.         % The user can input that using two formats: points that he wants to get the 1st derivative
33.         % or the boundaries(a,b) of an period[a,b] besides a number of periods(n).
34.     while 1
35.         disp('Please Choose points Input mode (Points or Boundary)');
36.         disp('Points like [X0 X1 X2 ... ]');
37.         disp('Boundary means enter [a,b] and n');
38.         Point_Mode = input('','s');
39.         disp('-------------------------------------');
40.         if ~(strcmpi( Point_Mode , 'Boundary' )||strcmpi( Point_Mode , 'Points' ))
41.             disp('Error!');
```

```
42.         else
43.             break;
44.         end;
45.     end;
46.     % Here, the code deals with the one of the two formats (Go back to Line 31).
47.     if strcmpi( Point_Mode , 'Boundary' )
48.         a = input('Please Enter [a,b]\n  a = ');
49.         b = input('  b = ');
50.         n = input('Please Enter number of periods\n  n = ');
51.         h = (abs(b-a))/n;
52.         X = a:h:b;
53.         h(1:n+1) = h(1);
54.     else
55.         X = input('Please Enter X \n For Example X = [0,1,5,6]\n  X = ');
56.         % Enter The Value of Step Size h.
57.         disp('Enter a value of (h) OR the corresponding (h) for every (x)');
58.         disp('Note: if numbers of (h) is less than number of (x), the remaining');
59.         disp('corresponding values of (h) will be last (h) entered');
60.         disp('------------------------------------');
61.         h = input('Please Enter the value of Step Size\n  h = ');
62.         n = numel(X);
63.         nh = numel(h);
64.         % If data not complate we let the other h equal last value of h.
65.         if nh<n
66.             h(nh:n) = h(nh);
67.         end;
68.     end;
69.     disp('------------------------------------');
70.
71.     % Find First Derivative.
72.     n = numel(X);
73.     FD = X;
74.     for i = 1:n
75.         FD(i) = (feval(f,X(i)+h(i))-feval(f,X(i)-h(i)))/(2*h(i));
76.     end;
77.
78.     %Display X, h, First Derivative.
79.     fprintf('f(x) = %s\n',fs);
80.     disp   ('+---------------+------------+-------------------+');
81.     fprintf('|      x        |      h     | first derivative  |\n');
82.     disp   ('+---------------+------------+-------------------+');
83.     for i = 1:n
84.         fprintf('| %12.4f  |  %8.4f  |   %12.4f    |\n',X(i),h(i),FD(i));
85.     end;
```

```
    disp   ('+---------------+------------+-------------------+');

end;

%% Data Mode
% The user has chose the Data mode (Go back to line 11 ).
if strcmpi( Mode , 'Data' )
     % The user has to enter the Boundary of interval.
     a = input('Please Enter the interval [a,b]\n  a = ');
     b = input('  b = ');
     n = input('Please Enter number of periods\n  n = ');
     if b<a
         tmp = a;
         a = b;
            b = tmp;
        end;
        h = (b-a)/n;
        X = a:h:b;

        % The user has to enter Y(x).
        Y = X;
        for i = 1:n+1
           fprintf('Please Enter Y(%1.4f) = ',X(i));
           Y(i) = input('');
        end;

        % Find First Derivative.
        FD = X;
        for i = 2:n
            FD(i) = (Y(i+1)-Y(i-1))/(2*h);
        end;

        %Display X, Y, First Derivative.
       disp   ('+---------------+---------------+---------------------+');
       fprintf('|      x         |      y         |   first derivative  |\n');
       disp   ('+---------------+---------------+---------------------+');
       fprintf('| %12.4f  | %12.4f  | Can`t be calculated |\n',X(1),Y(1));
       for i = 2:n
           fprintf('| %12.4f  | %12.4f  |     %12.4f    |\n',X(i),Y(i),FD(i));
       end;
       fprintf('| %12.4f  | %12.4f  | Can`t be calculated |\n',X(n+1),Y(n+1));
       disp   ('+---------------+---------------+---------------------+');

   end;
```

First Derivative-Forward Difference:

```
1.  %% Forward Difference
2.  % The first derivative " *Forward Difference* " metrhod is implemented as
3.  % shown here using MATLAB(R).
4.
5.  %% Clear Memory and Screen
6.  clear
7.  clc
8.
9.  %% Firstly, the user has to choose between two input modes
10. % 1- Function: Formula of a function.
11. % 2- Data    : some (x,y) points.
12. % the user has to write one of two words (Function or Data) to indicate the mode.
13. while 1
14.     disp('Please Choose your Input mode (Function or Data)');
15.     Mode = input('','s');
16.     disp('-------------------------------------');
17.     if ~(strcmpi( Mode , 'Function' )||strcmpi( Mode , 'Data' ))
18.         disp('Error!')
19.     else
20.         break;
21.     end;
22. end;
23.
24. %% Function Mode
25. % The user has chosen the Function mode (Go to line 10 ).
26. if strcmpi( Mode , 'Function' )
27.     fs = input('Please Enter The Function \n f(x)= ','s');
28.     f  = inline(fs);
29.     disp('-------------------------------------');
30.
31.     % Here, the user has to define where he wants to get the values of the 1st derivative.
32.         % The user can input that using two formats: points that he wants to get the 1st derivative
33.         % or the boundaries(a,b) of an period[a,b] besides a number of periods(n).
34.     while 1
35.         disp('Please Choose points Input mode (Points or Boundary)');
36.         disp('Points like [X0 X1 X2 ... ]');
37.         disp('Boundary means enter [a,b] and n');
38.         Point_Mode = input('','s');
39.         disp('-------------------------------------');
40.         if ~(strcmpi( Point_Mode , 'Boundary' )||strcmpi( Point_Mode , 'Points' ))
41.             disp('Error!');
```

```
42.         else
43.             break;
44.         end;
45.     end;
46.     % Here, the code deals with the one of the two formats (Go back to Line 31).
47.     if strcmpi( Point_Mode , 'Boundary' )
48.         a = input('Please Enter [a,b]\n  a = ');
49.         b = input('  b = ');
50.         n = input('Please Enter number of periods\n  n = ');
51.         h = (abs(b-a))/n;
52.         X = a:h:b;
53.         h(1:n+1) = h(1);
54.     else
55.         X = input('Please Enter X \n For Example X = [0,1,5,6]\n  X = ');
56.         % Enter The Value of Step Size h.
57.         disp('Enter a value of (h) OR the corresponding (h) for every (x)');
58.         disp('Note: if numbers of (h) is less than number of (x), the remaining');
59.         disp('corresponding values of (h) will be last (h) entered');
60.         disp('-------------------------------------');
61.         h = input('Please Enter the value of Step Size\n  h = ');
62.         n = numel(X);
63.         nh = numel(h);
64.         % If data not complate we let the other h equal last value of h.
65.         if nh<n
66.             h(nh:n) = h(nh);
67.         end;
68.     end;
69.     disp('-------------------------------------');
70.
71.     % Find First Derivative.
72.     n = numel(X);
73.     FD = X;
74.     for i = 1:n
75.         FD(i) = (feval(f,X(i)+h(i))-feval(f,X(i)))/(h(i));
76.     end;
77.
78.     %Display X, h, First Derivative.
79.     fprintf('f(x) = %s\n',fs);
80.     disp   ('+---------------+------------+-------------------+');
81.     fprintf('|      x        |      h     | first derivative  |\n');
82.     disp   ('+---------------+------------+-------------------+');
83.     for i = 1:n
84.         fprintf('| %12.4f  |  %8.4f  |   %12.4f    |\n',X(i),h(i),FD(i));
85.     end;
```

```
    disp   ('+---------------+------------+-------------------+');

end;

%% Data Mode
% The user has chose the Data mode (Go back to line 11 ).
if strcmpi( Mode , 'Data' )
     % The user has to enter the Boundary of interval.
     a = input('Please Enter the interval [a,b]\n  a = ');
     b = input('  b = ');
     n = input('Please Enter number of periods\n  n = ');
     if b<a
         tmp = a;
         a = b;
                b = tmp;
            end;
            h = (b-a)/n;
            X = a:h:b;

           % The user has to enter Y(x).
            Y = X;
            for i = 1:n+1
               fprintf('Please Enter Y(%1.4f) = ',X(i));
               Y(i) = input('');
            end;

            % Find First Derivative.
            FD = X;
            for i = 1:n
                FD(i) = (Y(i+1)-Y(i))/h;
            end;

            %Display X, Y, First Derivative.
           disp   ('+---------------+---------------+---------------------+');
           fprintf('|      x        |      y        |   first derivative  |\n');
           disp   ('+---------------+---------------+---------------------+');
           for i = 1:n
               fprintf('| %12.4f  | %12.4f  |     %12.4f    |\n',X(i),Y(i),FD(i));
           end;
           fprintf('| %12.4f  | %12.4f  | Can`t be calculated |\n',X(n+1),Y(n+1));
           disp   ('+---------------+---------------+---------------------+');


       end;
```

First Derivative-Backward Difference:

```
%% Backward Difference
% The first derivative " *% Backward Difference Difference* " metrhod is implemented as
% shown here using MATLAB(R).

%% Clear Memory and Screen
clear
clc

%% Firstly, the user has to choose between two input modes
% 1- Function: Formula of a function.
% 2- Data     : some (x,y) points.
% the user has to write one of two words (Function or Data) to indicate the mode.
while 1
    disp('Please Choose your Input mode (Function or Data)');
    Mode = input('','s');
    disp('-------------------------------------');
    if ~(strcmpi( Mode , 'Function' )||strcmpi( Mode , 'Data' ))
        disp('Error!')
    else
        break;
    end;
end;

%% Function Mode
% The user has chosen the Function mode (Go to line 10 ).
if strcmpi( Mode , 'Function' )
    fs = input('Please Enter The Function \n f(x)= ','s');
    f  = inline(fs);
    disp('-------------------------------------');

    % Here, the user has to define where he wants to get the values of the 1st derivative.
        % The user can input that using two formats: points that he wants to get the 1st derivative
        % or the boundaries(a,b) of an period[a,b] besides a number of periods(n).
    while 1
        disp('Please Choose points Input mode (Points or Boundary)');
        disp('Points like [X0 X1 X2 ... ]');
        disp('Boundary means enter [a,b] and n');
        Point_Mode = input('','s');
        disp('-------------------------------------');
        if ~(strcmpi( Point_Mode , 'Boundary' )||strcmpi( Point_Mode , 'Points' ))
            disp('Error!');
```

```
42.         else
43.             break;
44.         end;
45.     end;
46.     % Here, the code deals with the one of the two formats (Go back to Line 31).
47.     if strcmpi( Point_Mode , 'Boundary' )
48.         a = input('Please Enter [a,b]\n  a = ');
49.         b = input('  b = ');
50.         n = input('Please Enter number of periods\n  n = ');
51.         h = (abs(b-a))/n;
52.         X = a:h:b;
53.         h(1:n+1) = h(1);
54.     else
55.         X = input('Please Enter X \n For Example X = [0,1,5,6]\n  X = ');
56.         % Enter The Value of Step Size h.
57.         disp('Enter a value of (h) OR the corresponding (h) for every (x)');
58.         disp('Note: if numbers of (h) is less than number of (x), the remaining');
59.         disp('corresponding values of (h) will be last (h) entered');
60.         disp('-------------------------------------');
61.         h = input('Please Enter the value of Step Size\n  h = ');
62.         n = numel(X);
63.         nh = numel(h);
64.         % If data not complate we let the other h equal last value of h.
65.         if nh<n
66.             h(nh:n) = h(nh);
67.         end;
68.     end;
69.     disp('-------------------------------------');
70.
71.     % Find First Derivative.
72.     n = numel(X);
73.     FD = X;
74.     for i = 1:n
75.         FD(i) = (feval(f,X(i))-feval(f,X(i)-h(i)))/(h(i));
76.     end;
77.
78.     %Display X, h, First Derivative.
79.     fprintf('f(x) = %s\n',fs);
80.     disp   ('+---------------+------------+-------------------+');
81.     fprintf('|      x        |      h     | first derivative  |\n');
82.     disp   ('+---------------+------------+-------------------+');
83.     for i = 1:n
84.         fprintf('| %12.4f  |  %8.4f  |   %12.4f    |\n',X(i),h(i),FD(i));
85.     end;
```

```
    disp   ('+---------------+------------+-------------------+');

end;

%% Data Mode
% The user has chose the Data mode (Go back to line 11 ).
if strcmpi( Mode , 'Data' )
     % The user has to enter the Boundary of interval.
     a = input('Please Enter the interval [a,b]\n  a = ');
     b = input('  b = ');
     n = input('Please Enter number of periods\n  n = ');
     if b<a
         tmp = a;
         a = b;
               b = tmp;
         end;
         h = (b-a)/n;
         X = a:h:b;

         % The user has to enter Y(x).
         Y = X;
         for i = 1:n+1
            fprintf('Please Enter Y(%1.4f) = ',X(i));
            Y(i) = input('');
         end;

         % Find First Derivative.
         FD = X;
         for i = 2:n+1
             FD(i) = (Y(i)-Y(i-1))/(h);
         end;

         %Display X, Y, First Derivative.
        disp   ('+---------------+---------------+---------------------+');
        fprintf('|      x        |      y        |   first derivative  |\n');
        disp   ('+---------------+---------------+---------------------+');
        fprintf('| %12.4f  | %12.4f  | Can`t be calculated |\n',X(1),Y(1));
        for i = 2:n+1
            fprintf('| %12.4f  | %12.4f  |     %12.4f    |\n',X(i),Y(i),FD(i));
        end;
        disp   ('+---------------+---------------+---------------------+');


    end;
```

First To Sixth Order Derivative-Central Difference:

```
% Central Difference
% The Higher derivative Up to 6 " *Central Difference* " metrhod is implemented as
% shown here using MATLAB(R).

%% Clear Memory and Screen
clear
clc

%% Firstly, the user has to choose between two input modes
% 1- Function: Formula of a function.
% 2- Data    : some (x,y) points.
% the user has to write one of two words (Function or Data) to indicate the mode.
while 1
    disp('Please Choose your Input mode (Function or Data)');
    Mode = input('','s');
    disp('-------------------------------------');

    if ~(strcmpi( Mode , 'Function' )||strcmpi( Mode , 'Data' ))
        disp('Error!')
    else
        break;
    end;
end;
%% Enter the order of Derivative
while 1
    disp('Please Enter The order of Derivative (1 to 6)');
    Order = input('  Order = ');
    disp('-------------------------------------');
    if (Order ~= 1)&&(Order ~= 2)&&(Order ~= 3)&&(Order ~= 4)&&(Order ~= 5)&&(Order ~= 6)
        disp('Error!');
    else
        break;
    end;
end;

%% Function Mode
% The user has chosen the Function mode (Go to line 10 ).
if strcmpi( Mode , 'Function' )
    % Enter the function
    fs = input('Please Enter The Function \n f(x)= ','s');
    f  = inline(fs);
    disp('-------------------------------------');
```

```
43.
44.     % Here, the user has to define where he wants to get the values of the 1st derivative.
45.         % The user can input that using two formats: points that he wants to get the 1st derivative
46.         % or the boundaries(a,b) of an period[a,b] besides a number of periods(n).
47.     while 1
48.         disp('Please Choose points Input mode (Points or Boundary)');
49.         disp('Points like [X0 X1 X2 ... ]');
50.         disp('Boundary means enter [a,b] and n');
51.         Point_Mode = input('','s');
52.         disp('-------------------------------------');
53.         if ~(strcmpi( Point_Mode , 'Boundary' )||strcmpi( Point_Mode , 'Points' ))
54.             disp('Error!');
55.         else
56.             break;
57.         end;
58.     end;
59.
60.     % Here, the code deals with the one of the two formats (Go back to Line 44).
61.     if strcmpi( Point_Mode , 'Boundary' )
62.         a = input('Please Enter [a,b]\n  a = ');
63.         b = input('  b = ');
64.         n = input('Please Enter number of periods\n  n = ');
65.         h = (abs(b-a))/n;
66.         X = a:h:b;
67.         h(1:n+1) = h(1);
68.     else
69.         X = input('Please Enter X \n For Example X = [0,1,5,6]\n  X = ');
70.         %Enter The Value of Step Size h
71.         disp('Enter a value of (h) OR the corresponding (h) for every (x)');
72.         disp('Note: if numbers of (h) is less than number of (x), the remaining');
73.         disp('corresponding values of (h) will be last (h) entered');
74.         disp('-------------------------------------');
75.         h = input('Please Enter the value of Step Size\n  h = ');
76.         n = numel(X);
77.         nh = numel(h);
78.         % If data not complate for unequal we let the other h equal last value of h
79.         if nh<n
80.             h(nh:n) = h(nh);
81.         end;
82.     end;
83.     disp('-------------------------------------');
84.
85.     % Find Needed Derivative.
```

```
86.     n = numel(X);
87.     HD = X;
88.     for i = 1:n
89.         if Order == 1
90.             HD(i) = (feval(f,X(i)+h(i))-feval(f,X(i)-h(i)))/(2*h(i));
91.         elseif Order == 2
92.             HD(i) = (feval(f,X(i)+h(i))+feval(f,X(i)-h(i))-2*feval(f,X(i)))/(h(i)*h(i));
93.         elseif Order == 3
94.             HD(i) = (feval(f,X(i)+2*h(i))-2*feval(f,X(i)+h(i))+2*feval(f,X(i)-h(i))-feval(f,X(i)-2*h(i)))/(2*h(i)^3);
95.         elseif Order == 4
96.             HD(i) = (feval(f,X(i)+2*h(i))-4*feval(f,X(i)+h(i))+6*feval(f,X(i))-4*feval(f,X(i)-h(i))+feval(f,X(i)-2*h(i)))/(h(i)^4);
97.         elseif Order == 5
98.             HD(i) = (feval(f,X(i)+3*h(i))-4*feval(f,X(i)+2*h(i))+5*feval(f,X(i)+h(i))-5*feval(f,X(i)-h(i))+4*feval(f,X(i)-2*h(i))-feval(f,X(i)-3*h(i)))/(2*h(i)^5);
99.         elseif Order == 6
100.                    HD(i) = (feval(f,X(i)+3*h(i))-6*feval(f,X(i)+2*h(i))+15*feval(f,X(i)+h(i))-20*feval(f,X(i))+15*feval(f,X(i)-h(i))-6*feval(f,X(i)-2*h(i))+feval(f,X(i)-3*h(i)))/(h(i)^6);
101.                end;
102.            end;
103.
104.            %Display X, h, Needed Derivative.
105.            fprintf('f(x) = %s\n',fs);
106.            disp   ('+---------------+------------+------------------+');
107.            fprintf('|      x         |      h      |    %1.0f derivative    |\n',Order);
108.            disp   ('+---------------+------------+------------------+');
109.            for i = 1:n
110.                fprintf('| %12.4f  |  %8.4f  |   %12.4f    |\n',X(i),h(i),HD(i));
111.            end;
112.            disp   ('+---------------+------------+------------------+');
113.        end;
114.
115.        %% Data Mode
116.        % The user has chose the Data mode (Go back to line 11 ).
117.        if strcmpi( Mode , 'Data' )
118.            % The user has to enter the Boundary of interval.
119.            a = input('Please Enter the interval [a,b]\n  a = ');
120.            b = input('  b = ');
121.            n = input('Please Enter number of periods\n  n = ');
122.            if b<a
123.                tmp = a;
124.                a = b;
125.                b = tmp;
```

```
        end;
        h = (b-a)/n;
        X = a:h:b;

        % The user has to enter Y(x).
        Y = X;
        for i = 1:n+1
           fprintf('Please Enter Y(%1.4f) = ',X(i));
           Y(i) = input('');
        end;

        % Find Needed Derivative.
        n = numel(X);
        HD = X;
        HP = floor((Order+1)/2); % half of points that we need to find the drivative
        for i = HP+1 : n-HP
            if Order == 1
                HD(i) = (Y(i+1)-Y(i-1))/(2*h);
            elseif Order == 2
                HD(i) = (Y(i+1)+Y(i-1)-2*Y(i))/(h*h);
            elseif Order == 3
                HD(i) = (Y(i+2)-2*Y(i+1)+2*Y(i-1)-Y(i-2))/(2*h^3);
            elseif Order == 4
                HD(i) = (Y(i+2)-4*Y(i+1)+6*Y(i)-4*Y(i-1)+Y(i-2))/(h^4);
            elseif Order == 5
                HD(i) = (Y(i+3)-4*Y(i+2)+5*Y(i+1)-5*Y(i-1)+4*Y(i-2)-Y(i-3))/(2*h^5);
            elseif Order == 6
                HD(i) = (Y(i+3)-6*Y(i+2)+15*Y(i+1)-20*Y(i)+15*Y(i-1)-6*Y(i-2)+Y(i-3))/(h^6);
            end;
        end;

        %Display X, Y, H, HD
        disp   ('+---------------+---------------+---------------------+');
        fprintf('|      x        |      y        |     %1.0f derivative    |\n',Order);
        disp   ('+---------------+---------------+---------------------+');
        for i = 1 : n
            if (i<HP+1)||(i>n-HP)
                fprintf('| %12.4f  | %12.4f  | Can`t be calculated |\n',X(i),Y(i));
            else
                fprintf('| %12.4f  | %12.4f  |     %12.4f    |\n',X(i),Y(i),HD(i));
            end;
        end;
        disp   ('+---------------+---------------+---------------------+');
    end;
```

Higher Order Derivative-Central Difference:

```
1.  %% Central Difference
2.  % The Higher derivative Up to n " *Central Difference* " metrhod is implemented as
3.  % shown here using MATLAB(R).
4.
5.  %% Clear Memory and Screen
6.  clear
7.  clc
8.
9.  %% Here, The user enter the order of derivative that he needs.
10. disp('Please Enter The order of Derivative');
11. Order = input('  Order = ');
12. disp('-------------------------------------');
13.
14. %% Enter The Function.
15. fs = input('Please Enter The Function \n f(x)= ','s');
16. f  = inline(fs);
17.
18. %% Input Mode
19. % Here, the user has to define where he wants to get the values of the 1st derivative.
20. % The user can input that using two formats: points that he wants to get the 1st derivative
21. % or the boundaries(a,b) of an period[a,b] besides a number of periods(n).
22. disp('-------------------------------------');
23. while 1
24.     disp('Please Choose points Input mode (Points or Boundary)');
25.     disp('Points like [X0 X1 X2 ... ]');
26.     disp('Boundary means enter [a,b] and n');
27.     Point_Mode = input('','s');
28.     disp('-------------------------------------');
29.     if ~(strcmpi( Point_Mode , 'Boundary' )||strcmpi( Point_Mode , 'Points' ))
30.         disp('Error!');
31.     else
32.         break;
33.     end;
34. end;
35. % Here, the code deals with the one of the two formats (Go back to Line 19).
36. if strcmpi( Point_Mode , 'Boundary' )
37.     a = input('Please Enter [a,b]\n  a = ');
38.     b = input('  b = ');
39.     n = input('Please Enter number of periods\n  n = ');
40.     h = (abs(b-a))/n;
41.     X = a:h:b;
42.     h(1:n+1) = h(1);
```

```
else
    X = input('Please Enter X \n For Example X = [0,1,5,6]\n  X = ');
    %Enter The Value of Step Size h.
    disp('Enter a value of (h) OR the corresponding (h) for every (x)');
    disp('Note: if numbers of (h) is less than number of (x), the remaining');
    disp('corresponding values of (h) will be last (h) entered');
    disp('-------------------------------------');
    h = input('Please Enter the value of Step Size\n  h = ');
    n = numel(X);
    nh = numel(h);
    % If data not complate for unequal we let the other h equal last value of h.
    if nh<n
        h(nh:n) = h(nh);
    end;
end;
disp('-------------------------------------');

% Find the Needed Derivative.
n = numel(X);
HD = X;
for j = 1:n
    HD(j) = 0;
    for i = 0:Order
        HD(j) = HD(j) + ((-1)^i) * nchoosek(Order,i) * feval(f,X(j)+(((Order/2)-i)*h(j)));
    end;
    HD(j) = HD(j)/(h(j)^Order);
end;

%Display X, h,  HD
fprintf('f(x) = %s\n',fs);
disp   ('+---------------+------------+-------------------+');
fprintf('|      x         |      h      |    %1.0f derivative    |\n',Order);
disp   ('+---------------+------------+-------------------+');
for i = 1:n
    fprintf('| %12.4f  |  %8.4f  |   %12.4f    |\n',X(i),h(i),HD(i));
end;
disp   ('+---------------+------------+-------------------+');
```

Richardson’s Extrapolation:

```
%% Richardson Extrapolation
% The first derivative " *Richardson Extrapolation* " method is implemented as
% shown here using MATLAB(R).

%% Clear Momory and Screen
clear
clc
%% Enter The Function
fs = input('Please Enter The Function \n f(x)= ','s');
f  = inline(fs);
disp('-------------------------------------');

%% Choose Output Mode
% 1- Direvative table (Dn1, Dn2, Dn3, ....) for only one point.
% 2- only the value of Derivative for many value of x.
while 1
    disp('choose output mode');
    disp('1 => Direvative table (Dn1, Dn2, Dn3, ....) for only one point');
    disp('2 => only the value of Derivative for many value of x');
    Out_Mode = input('Enter mode (1 or 2) = ');
    disp('-------------------------------------');
    if Out_Mode == 1 || Out_Mode == 2
        break;
    else
        disp('Error');
    end;
end;

%% Input Mode
% Here, the user has to define where he wants to get the values of the 1st derivative.
% The user can input that using two formats: points that he wants to get the 1st derivative
% or the boundaries(a,b) of an period[a,b] besides a number of periods(n).

% 1- Direvative Table Mode.
if Out_Mode == 1
    Point_Mode = 'Points'; %Just one Point X(1).
end;
% 2- Value Mode.
while Out_Mode == 2
        disp('Please Choose points Input mode (Points or Boundary)');
        disp('Points like [X0 X1 X2 ... ]');
        disp('Boundary means enter [a,b] and n');
```

```
        Point_Mode = input('','s');
        disp('-------------------------------------');
        if ~(strcmpi( Point_Mode , 'Boundary' )||strcmpi( Point_Mode , 'Points' ))
        disp('Error!');
    else
        break;
        end;
end;

%% Enter Boundary or Points :: Values of X
if strcmpi( Point_Mode , 'Boundary' )
        a = input('Please Enter [a,b]\n  a = ');
        b = input('  b = ');
    n = input('Please Enter number of periods\n  n = ');
    h = (abs(b-a))/n;
    X = a:h:b;
    h(1:n+1) = h(1);
else
    X = input('Please Enter the value of x = ');
end;
disp('-------------------------------------');

%% Choose the accuracy (h,m)
disp('Choose the accuracy you need');
disp('If the T.E = O(h^m) and h<1 then for greater m we get small error');
while 1
    disp('Note: m must be even number');
    m = input('Enter m = ');
    disp('-------------------------------------');
    if (mod(m,2) == 0)
        break;
    end;
end;
m = int32(m/2);

%% Enter The Value of Step Size (h) for points mode
if strcmpi( Point_Mode , 'Points' )
    if Out_Mode == 2
        disp('Enter a value of (h) OR the corresponding (h) for every (x)');
        disp('Note: if numbers of (h) is less than number of (x), the remaining');
        disp('corresponding values of (h) will be last (h) entered');
        disp('-------------------------------------');
    end;
    h = input('Please Enter the value of Step Size\n  h = ');
```

```
    disp('-------------------------------------');
    n = numel(X);
    nh = numel(h);
    % If data not complate for unequal we let the other h equal last value of h.
    if nh<n
        h(nh:n) = h(nh);
    end;
end;

%% Find First Drivative

D = zeros(m); % prepare size of the matrix D to avoid resizing.
% First Derivative of X(j) ==> FD
    FD = X; % prepare size of the matrix FD to avoid resizing.

    % Direvative Table Mode just use X(1).
    % This Condition to deal with more than one value of X.
    if Out_Mode == 1
        n = 1;
    else
        n = numel(X);
    end;

    for j = 1:n
        hj = h(j);
        D(1,1) = (feval(f,X(j)+hj)-feval(f,X(j)-hj))/(2*hj);
        for i = 1:1:m-1
            hj = hj/2;
            D(i+1,1) = (feval(f,X(j)+hj)-feval(f,X(j)-hj))/(2*hj);
            for k = 1:1:i
                D(i+1,k+1) = D(i+1,k)+(D(i+1,k)-D(i,k))/((4.0^double(k))-1);
            end;
        end;
        FD(j) = D(m,m);
    end;

    %% Output for Table Mode
    if Out_Mode == 1
        fprintf('f(x) = %s\n',fs);
        Line = '+------+------------+';
        for i = 1:m
            Line = strcat(Line,'-----------------+');
        end;

```

```
        disp(Line); % Before Header.
        fprintf('|   i  |      h     |');
        for i = 1:m
            fprintf('       Di,%1.0d      |',i);
        end;
        fprintf('\n');

        disp(Line); % Before Data.
        for i = 1:m
            fprintf('|  %2.0f  |  %8.4f  |',i,h);
            for j = 1:m
                if j>i
                    fprintf('                 |');
                else
                    fprintf('   %12.4f  |', D(i,j));
                end;
            end;
            fprintf('\n');
            h =h/2;
        end;

        disp(Line); % End.
    end;

    %% Output for Value Mode
    if Out_Mode == 2
        fprintf('f(x) = %s\n',fs);
        disp   ('+---------------+------------+-------------------+');
        fprintf('|      x        |      h     | first derivative  |\n');
        disp   ('+---------------+------------+-------------------+');
        for i = 1:n
            fprintf('| %12.4f  |  %8.4f  |   %12.4f    |\n',X(i),h(i),FD(i));
        end;
        disp   ('+---------------+------------+-------------------+');
    end;
```

References:

College Library:

1- Applied Numerical Methods for Engineering.
2- Numerical Analysis.
3- Computer Based Numerical And Statistical Techniques.

Wikipedia:

1- http://en.wikipedia.org/wiki/Numerical_differentiation
2- http://en.wikipedia.org/wiki/Finite_difference

Numerical Methods for Engineers, Dr.Waheed K Zahra.